

Performing Safety Evaluation on Detailed Hardware Level according to ISO 26262

Nico Adler and Stefan Otten
FZI Research Center for Information Tech

Philippe Cuenot
Continental Automotive France

Klaus Müller-Glaser
Karlsruhe Institute of Technology

ABSTRACT

Electronic design on detailed hardware level for automotive safety-related systems requires evaluation of the hardware architecture to cope with random hardware failures. The international standard ISO 26262 - functional safety for road vehicles - claims two methods: hardware architectural metrics and evaluation of safety goal violations as a probabilistic approach. Although the utilization of these analyses is required, annotations of failure data in combination with performing evaluation of a preliminary hardware architecture using deposited failure data is not supported in an integrated model-based development environment.

To overcome these inconveniences, we analyzed the ISO 26262, in particular Part 5 for product development at the hardware level, to provide both, meta-model for failure description of detailed hardware and performing evaluation of the hardware architecture. This UML-compliant meta-model expands existing EAST-ADL2 constructs. We implemented our concepts in a model-based architecture description language for large scaled electric and electronic architectures. To deposit specific failure rates and modes, we provide a library concept for hardware component types. Besides modeling schematics using the type library, an automatic preparation for characterization of safety-related hardware components is provided. To perform quantified analysis, we acquire relevant failure data using model queries and a metric framework to implement the equations for hardware evaluation according to ISO 26262 Part 5. Detailed results and compliance with target values including dedicated measures are documented in generated reports.

CITATION: Adler, N., Otten, S., Cuenot, P. and Müller-Glaser, K., "Performing Safety Evaluation on Detailed Hardware Level according to ISO 26262," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(1):2013, doi:10.4271/2013-01-0182.

1. INTRODUCTION

ISO 26262 - Road vehicles: Functional safety - [1], published in 2011, is an international standard, which adapts the safety standard IEC 61508 [2] for the automotive domain. This standard is structured in ten parts, as shown in [Figure 1](#). ISO 26262 Part 5 - 'Product development at the hardware level' [1] deals with the design of safety-related hardware and the corresponding development processes.

ISO 26262 Part 5 Clause 8 [1] describes the evaluation of the hardware architecture to cope with random hardware failures. It demands the application of architectural metrics on detailed hardware level. Therefore, two architectural metrics are provided: *single-point fault metric* and *latent-fault*

metric. These metrics deliver quantitative values for the verification of the overall robustness of the hardware design.

Part 5 Clause 9 [1] proposes two alternatives to evaluate the residual risk of safety goal violation as a complement to the hardware architectural metrics. The first method is called *Probabilistic Metric for random Hardware Failures* (PMHF), which is an overall probabilistic approach for the maximum probability of safety goal violation. The second method describes an individual categorization of each violating residual, single-point and multiple-point fault into *failure rate classes* (FRC) including a diagnostic coverage on hardware component level.

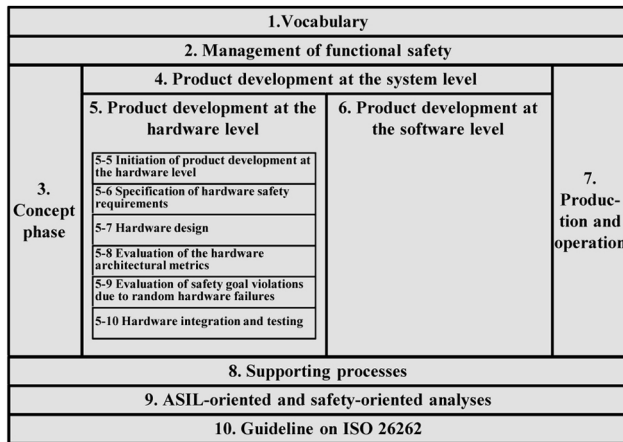


Figure 1. ISO 26262, focused on Part 5 - modified overview of [1]

A challenge for embedded systems is to provide failure resilience and reliability after integration, particularly influenced by the hardware architecture [3]. Main objective is to fulfill all safety requirements, in this context focusing on hardware. As complexity of hardware design continuously increases, designers have to be supported in the design process.

State of the art, numerous tools and research activities focus on evaluation of hardware in context of functional safety. However, there is actually no methodology which allows the combination of the following:

- Specification of failure data based on an electric and electronic architecture description language
- Annotation techniques for electric/electronic architecture specific content and failure data
- Modeling of detailed hardware design
- Performing checks and hardware evaluation in context of functional safety
- Integration of the described tasks in a single consistent user interface

The utilization of these topics would support traceability, iterative hardware design and accelerate development activities for functional safety, as the hardware design including safety evaluation is being integrated in an architecture description language. Additionally, it supports the documentation of the safety case claimed by ISO 26262.

This paper presents the outlined methodology and is structured as follows. The next section describes related work as well as basics of architecture description languages and failure and reliability in context of hardware. Section 3 presents our developed meta-model for hardware failure data, derived from the requirements of ISO 26262 Part 5 [1]. Section 4 provides a detailed conceptual description of evaluation of hardware architectures according to Part 5 Clause 8 and 9 [1]. In Section 5, the integration and

implementation of our concepts into a model-based architecture description language is demonstrated. Section 6 summarizes the results. The final section concludes the paper and gives an outlook on future work.

2. RELATED WORK

QuantUM [4] is a UML-based approach and describes support of the design process for especially automotive safety-related systems using semi-formal and formal methods. It is stated, that the QuantUM approach is not applicable for the description of hardware architectures. In principle, the QuantUM tool allows performing calculations of probabilistic metrics.

The work of Jeon et al. [5], published before publication of the ISO 26262, focuses superficially on the process of performing hardware architectural metrics. However, there is neither a semi-formal or formal method nor a prototype implementation.

In the article of Bellotti et al. [6] the evaluation of hardware is mentioned in the context of microprocessors and microcontrollers. It gives an overview about the impact of functional safety requirements for the design of microprocessors. Safety analysis of a proposed brake-by-wire system from a functional safety point of view is described in the article of Sinha [7].

The approach of Svancara et al. [8] provides a detailed description of Part 5 Clause 9 [1], particularly the second method for evaluation of safety goal violation, in context of an electrical powered steering. It analyzes the evaluation methods but provides no implementation or integration into a model-based environment.

The work of Papadopoulos, Walker et al. [9] describes failure analysis of an automotive system by linking architecture artifacts of EAST-ADL2 to the safety analysis tool HiP-HOPS [10]. This methodology allows performing failure mode and effect analysis (FMEA) and the automated generation of fault trees and analysis of them (FTA). Additionally, it facilitates the automatic allocation of safety integrity levels (SIL) [11].

A model-based approach for automotive safety-related systems and the extension of both, EAST-ADL2 and AUTOSAR, is current work in the European project *Safe Automotive soFtware architEcture* (SAFE) [12], organized in the framework of ITEA2. Within this project, the topics presented in this paper were partially addressed.

2.1. Architecture Description Languages

Different approaches address electric and electronic architecture modeling in context of automotive. AUTOSAR [13] describes a standardized automotive software architecture. It provides a basic infrastructure to design an automotive software system based on components. EAST-ADL2 [14] is an architecture description language developed in European research projects. It allows the description of software and electronics on different layers of abstraction,

based on concepts of UML, SysML and Architecture Analysis and Design Language (AADL) [15]. It is meant to complement AUTOSAR description on a higher level of abstraction.

Another model-based approach for the description of EEAs is the 'Electric Electronic Architecture - Analysis Design Language' (EEA-ADL) [16] [17]. This data model, related to EAST-ADL2, also forms the basis of the architecture modeling and analysis tool PREEvision [18] [19]. It provides different abstraction layers, as shown in Figure 2. *Product goals* consisting of requirements with customer features and feature functionality network constitute the first abstraction layer. Artifacts of this layer are text elements, which represent atomic features or requirements for the architecture. According to ISO 26262 safety goals (SG) with their specific automotive safety integrity level (ASIL) and both functional and technical safety requirements can be deposited and traced.

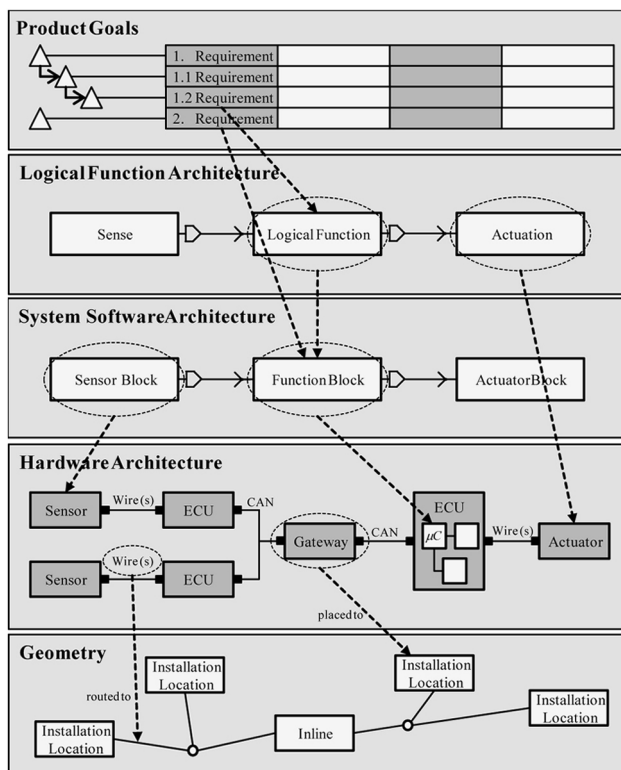


Figure 2. EEA layered architecture - modified overview of [18]

The underlying layers are: *logical function architecture*, *system software architecture*, *hardware architecture* consisting of component architecture with network topology, electric circuit and wiring harness, and the *geometry*. Cross layer links between model artifacts can be modeled using mappings. Apart from modeling EE relevant content, the EEA-ADL provides the opportunity to deposit attributes. This makes the model suitable for applying metrics and performing architecture evaluations based on the integrated and graphically notated metric framework. To build up model

queries based on the EEA data model, an integrated rule model is provided.

2.2. Failure and Reliability in context of Hardware

Reliability in context of hardware design is based on failure characteristics of single hardware components, which are specified in this chapter.

Failure Rate

Each hardware component, like a resistor, capacitor or complex component, has a failure rate λ_{HWComp} , which states how often the component fails in a certain time. Commonly, the failure rate is expressed in *Failure-In-Time* (FIT). 1 FIT is one failure in 10^9 h.

Failure Mode

The behavior of hardware components in case of a failure is described as failure modes. A component can have several failure modes, e.g. for a resistor common failure modes are short circuit and open circuit. For each failure mode, a failure rate distribution is specified. The failure rate distribution defines the ratio of the overall failure rate of a hardware component in percent that is distributed to the specific failure mode.

Hardware specific failure rates and failure modes can be taken from commonly recognized industry sources, statistical data (field or test) or expert judgment. Regarding industry sources, the ISO 26262 exemplarily recommends IEC/TR 62380 [20], IEC 61709:2011, UTE C80-811, MIL HDBK 338 [21]. Additionally, Siemens Handbook SN 29500 and FIDES guide 2009 [22] are advisable. In order to normalize failure rates taken from different sources, scaling factors for PMHF and FRC can be applied according to ISO 26262 Part 5 Annex F [1].

Fault Types

If a hardware component fails with a specific failure mode, this fault can have several impacts on the behavior of the whole item, considering a specific safety goal. This is expressed by specifying faults into different categories, as shown in Figure 3.

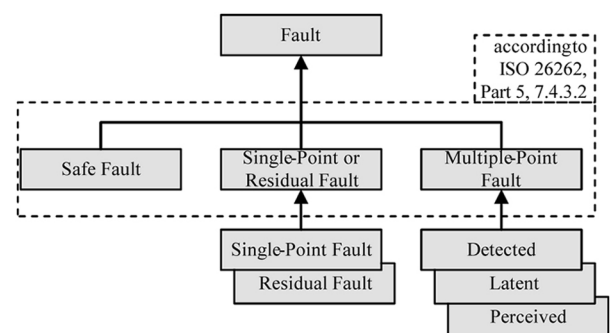


Figure 3. Fault types of hardware

Therefore, faults are primarily categorized by the potential violation of the considered safety goal. If the fault has no impact on the violation of the safety goal, it is specified as a *safe fault*. *Single-point or residual faults* directly lead to the violation of the safety goal, representing a fault of first order. Faults which lead to the violation of the safety goal in combination with other independent faults are categorized as *multiple-point faults*. These represent faults of higher order.

Going into detail, a *single-point fault* directly leads to a violation, as there is no safety mechanism present. If a safety mechanism covers the direct violation, the proportion of the fault which is not covered is the *residual fault*. *Multiple-point faults* can either be *detected* by a safety mechanism, *perceived* by the driver (through different system behavior) or *latent* (not perceived or detected).

Safety Mechanism

An implemented safety mechanism shall prevent single-point faults, reduce residual faults or prevent multiple-point faults from being latent. A safety mechanism has a diagnostic coverage KDC, which describes the effectiveness of the implemented safety mechanism. The diagnostic coverage is considered with respect to residual faults or latent multiple-point faults and expressed in percent.

Safety Goal

A safety goal defines a top-level safety requirement including an automotive safety integrity level (ASIL), with ASIL-D representing the most stringent and ASIL-A the least stringent level according to ISO 26262 Part 3 [1].

3. META-MODEL FOR HW FAILURE

In terms of ISO 26262 Part 5 [1], architectural metrics and evaluation of safety goal violations have to be performed to assess safety-related hardware designs. An architecture description language, according to Section 2, can serve as an integrated data model. For specific failure data we refer to the standards, presented in Section 2.2. Specific requirements to perform metrics at the detailed hardware level can be derived from ISO 26262 Part 5 [1].

To provide automated calculation of metrics, a meta-model for the description of detailed hardware failures and an implementation of the equations in a metric framework have to be provided. Based on the formalized meta-model the hardware designer shall be able to model his specific hardware design. Therefore, the meta-model shall provide all necessary data for the failure description of hardware components in a UML-compliant class diagram. Our meta-model is structured into three parts:

- *Meta-model for HW Failure* has to ensure that the failure data of hardware components can be annotated or associated.
- *Meta-model for HW Architectural Metrics* describes all classes and associations to perform the metric according to ISO 26262 Part 5 Clause 8 [1].

• *Meta-model for Probabilistic Approach* describes all classes and associations to perform the evaluation according to ISO 26262 Part 5 Clause 9 [1].

In this paper we present an excerpt of our meta-model for hardware failure, as this is the basic extension to existing approaches like EAST-ADL2. Our classes and their relations are presented in [Figure 4](#) and described in the following.

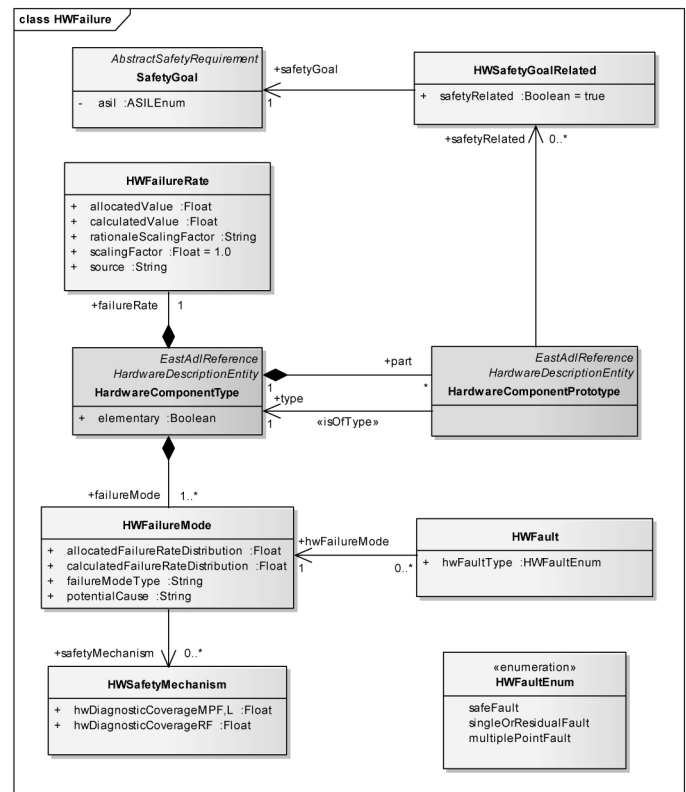


Figure 4. Concept: Excerpt of UML-compliant class diagram for meta-model hardware failure

The classes *HardwareComponentType* and *HardwareComponentPrototype* describe hardware components on detailed abstraction level, like resistors or capacitors. These classes and their relations reference the construct in EAST-ADL2 according to the domain model specification [14]. This construct facilitates the composition of several types of hardware components in an actual prototype, whereas the prototype itself is also representing a *HardwareComponentType*. Therefore, a *HardwareComponentType* can either be a single hardware component or a composition of these. The additional attribute *elementary* of type Boolean allows to describe an elementary hardware component.

The class ***HWFailureRate*** is composed of the class *HardwareComponentType* and shall allow the annotation of appropriate failure rate data for the hardware component. The composition has the multiplicity one due to the fact that each hardware component has a single failure rate description. The class provides the attributes ***CalculatedValue*** and